Session 118

ColorSync

Travis Brown

Session 118

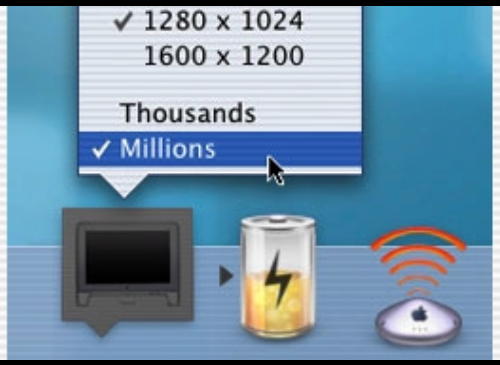
ColorSync on Mac OS X

David Hayward
Eric Broadbent
Luke Wallis
ColorSync Engineers

ColorSync on Mac OS X

- Overview of ColorSync
- ColorSync and Quartz
- Profiles and CMMs
- Preferences and Supporting APIs
- Device Integration
- Q&A

Overview of ColorSync



Overview of ColorSync

- Problem: Devices represent color in different spaces and different gamuts
- Solution: ColorSync is a complete color management system, designed to provide consistent color across devices
  - ColorSync can also be used to produce special color effects
- Based on:
  - ICC Profiles
  - Color Management Modules (CMMs)

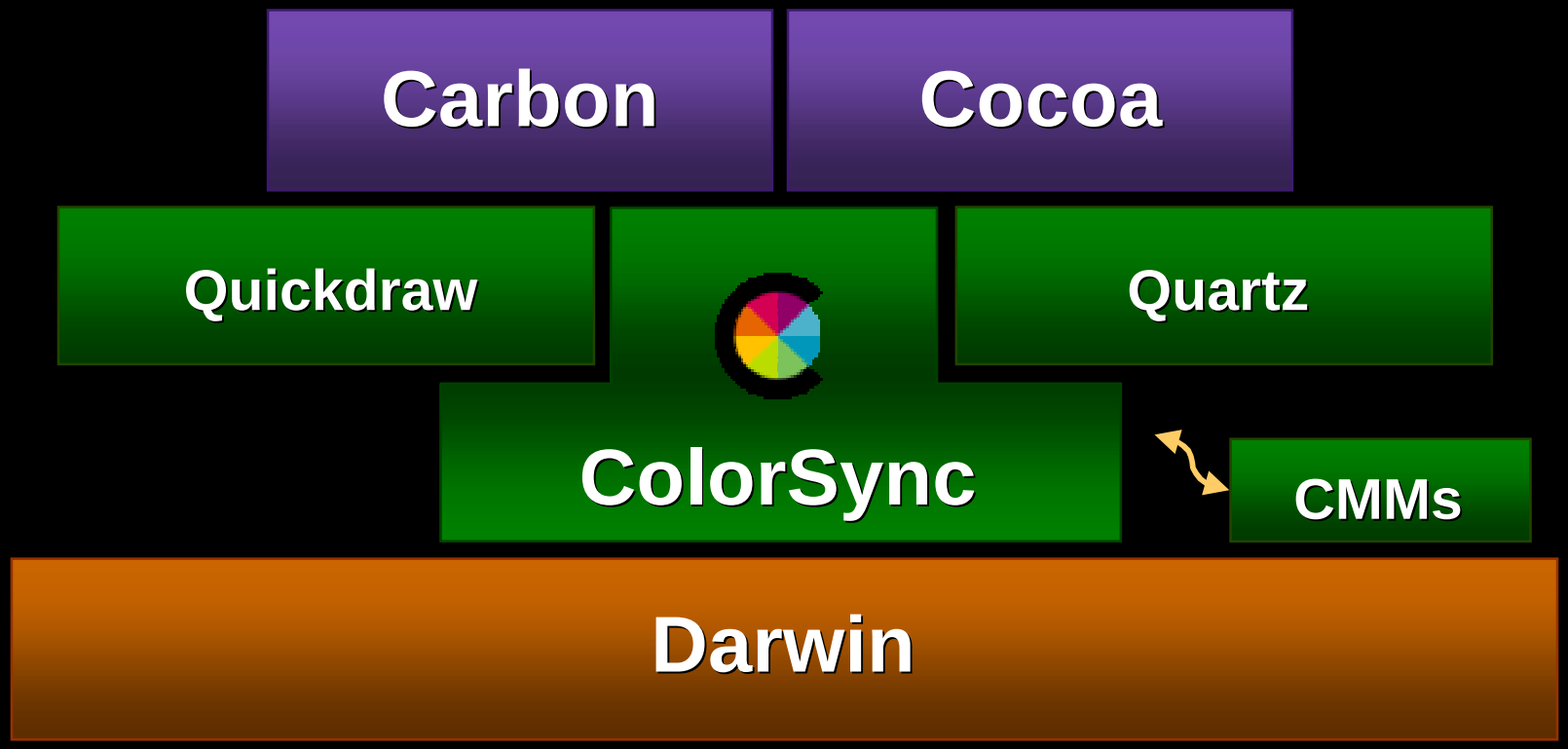
Overview of ColorSync

- ICC Profiles:
  - Cross platform file format defined by the International Color Consortium (www.color.org)
  - Profiles describe how to transform colors between device color space and interchange space (PCS)
  - Spec encompasses standard color spaces such as sRGB and NTSC
  - Processing model defined, but flexible and open
  - Ongoing improvements, active technical committees

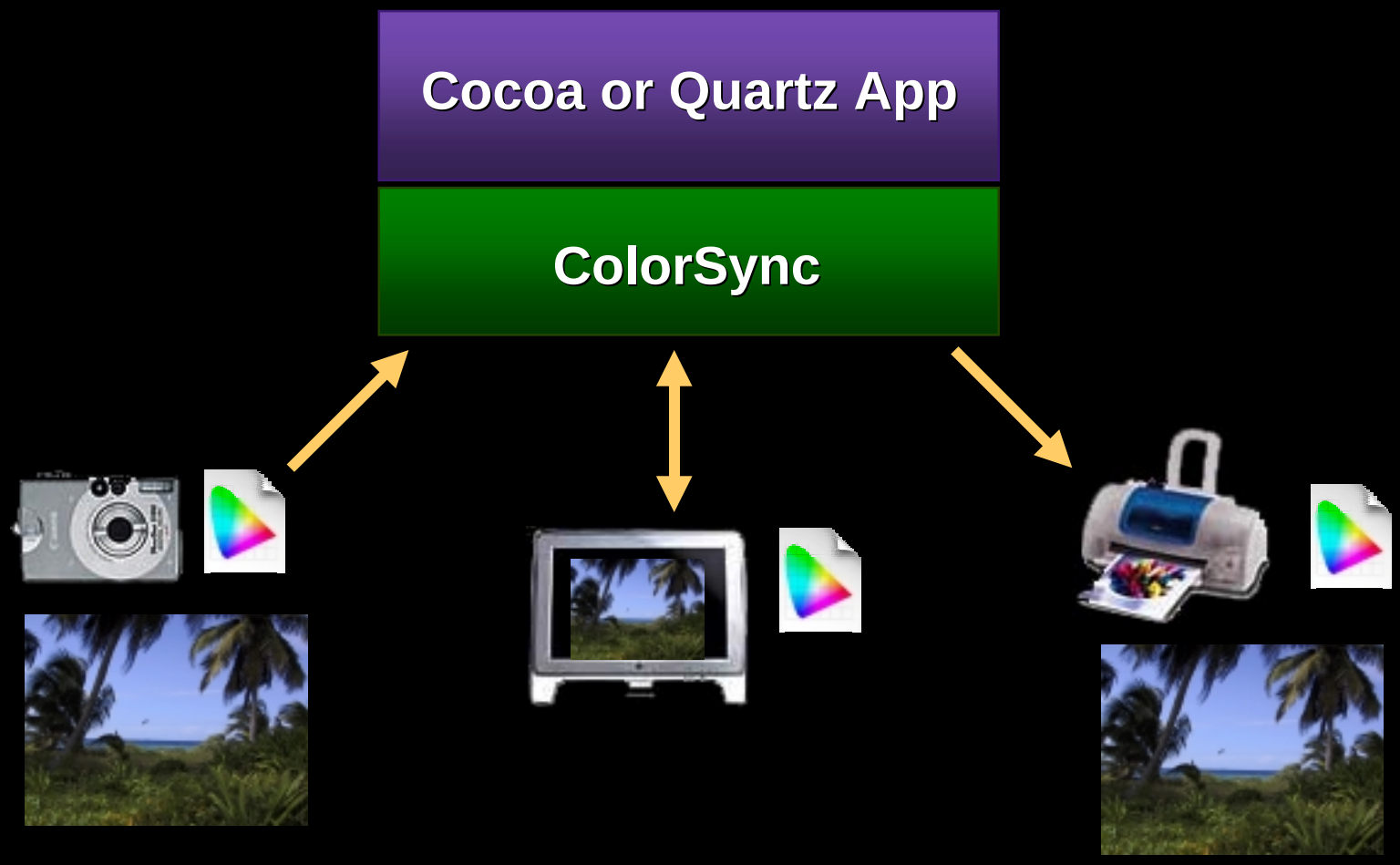
Overview of ColorSync

- Color Management Modules—CMMs:
  - Provide the mathematical engine to perform profile transformations
  - Apple CMM ships on Mac OS 9 and Mac OS X, is default CMM
  - Open system, 3rd-party CMMs available from Agfa, Kodak, Heidelberg, Imation, etc.

Overview of ColorSync



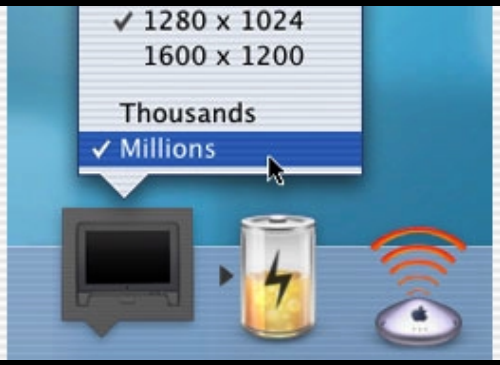
Overview of ColorSync



Overview of ColorSync

- ColorSync has evolved from 1.0 (1993) to 3.1 today
- ColorSync 2.x API on Mac OS 9 is 100% the same on Mac OS X
- ColorSync on Mac OS X has new features, APIs, and architectural changes which make it more powerful
- Quartz & Printing use ColorSync in Mac OS X 10.0

ColorSync and Quartz



ColorSync

and

Quartz Color Management

Quartz Color Management

- Objective
  - Integrated graphics and color management
- Requirements
  - Composite different color spaces and opacity
  - Scalable color management solution
  - Color accuracy
  - Performance
  - PDF compatibility

Quartz Color Management

PDF Color Model

+

ColorSync Engine

PDF Color Model

- PDF color definition
  - /DeviceGray, /DeviceRGB, /DeviceCMYK
  - /CalGray, /CalRGB, /CalLab
  - /ICCBased
- PDF color conversions

Function (source, destination, rendering intent)

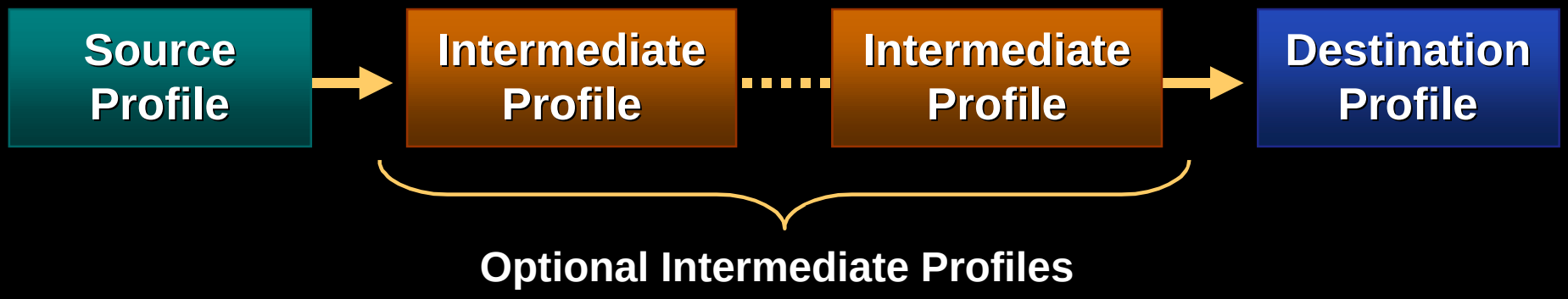
Source Color Space → **Destination Color Space**

Rendering intent = {perceptual, relative colorimetric, saturation, absolute colorimetric}

ColorSync/ICC Color Model

- ICC color definition
  - ICC profile
- ICC color conversions

$$\text{Function}\left(\sum_{i=2}^{n} \text{Profiles (rendering intent)}\right)$$



Rendering intent = {perceptual, relative colorimetric, saturation, absolute colorimetric}

Color Model in Quartz

Integration of PDF and ICC color models

- PDF color spaces expressed as ICC profiles
  - Device color spaces assigned default profiles

    /DeviceGray → Quartz Default Gray
    /DeviceRGB → Quartz Default RGB
    /DeviceCMYK → Quartz Default CMYK

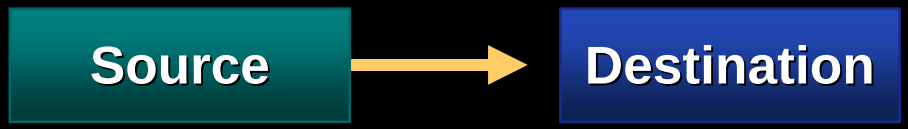
  - /CalGray, /CalRGB, /CalLab → ICC profiles
  - ICCBased color space provides an ICC profile

Color Model in Quartz

- PDF color space equivalence

If (source color space == destination color space)
{

no conversion is required – data remains unaltered

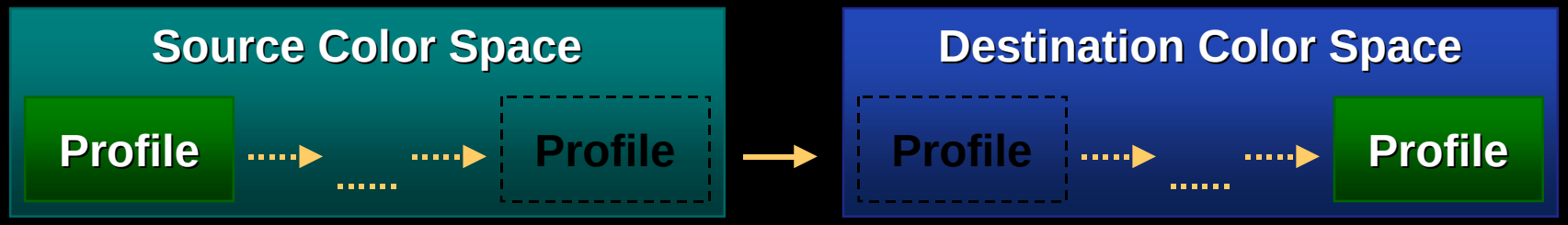


}
else
{

}

Color Model in Quartz

- Color space is one or more ICC profiles



  - Preserve PDF concept of matching source to destination
  - Suitable for advanced color management like softproofing, special effects, color device simulation, etc.
  - Can be concatenated by ColorSync to a single ICC profile

Color Model in Quartz

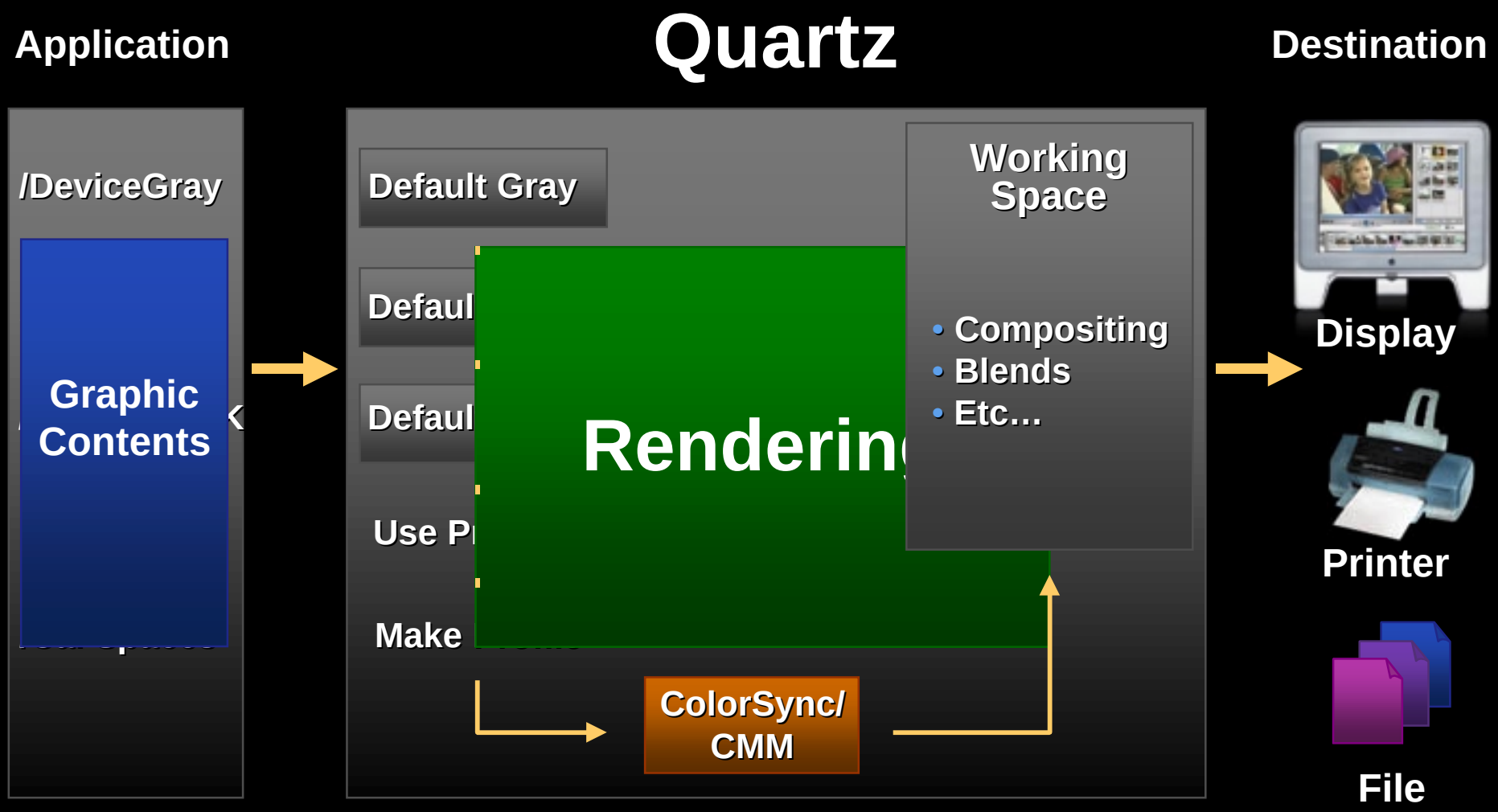
- Quartz implicit device color conversions

Quartz Default Gray = Gray to RGB → Quartz Default RGB

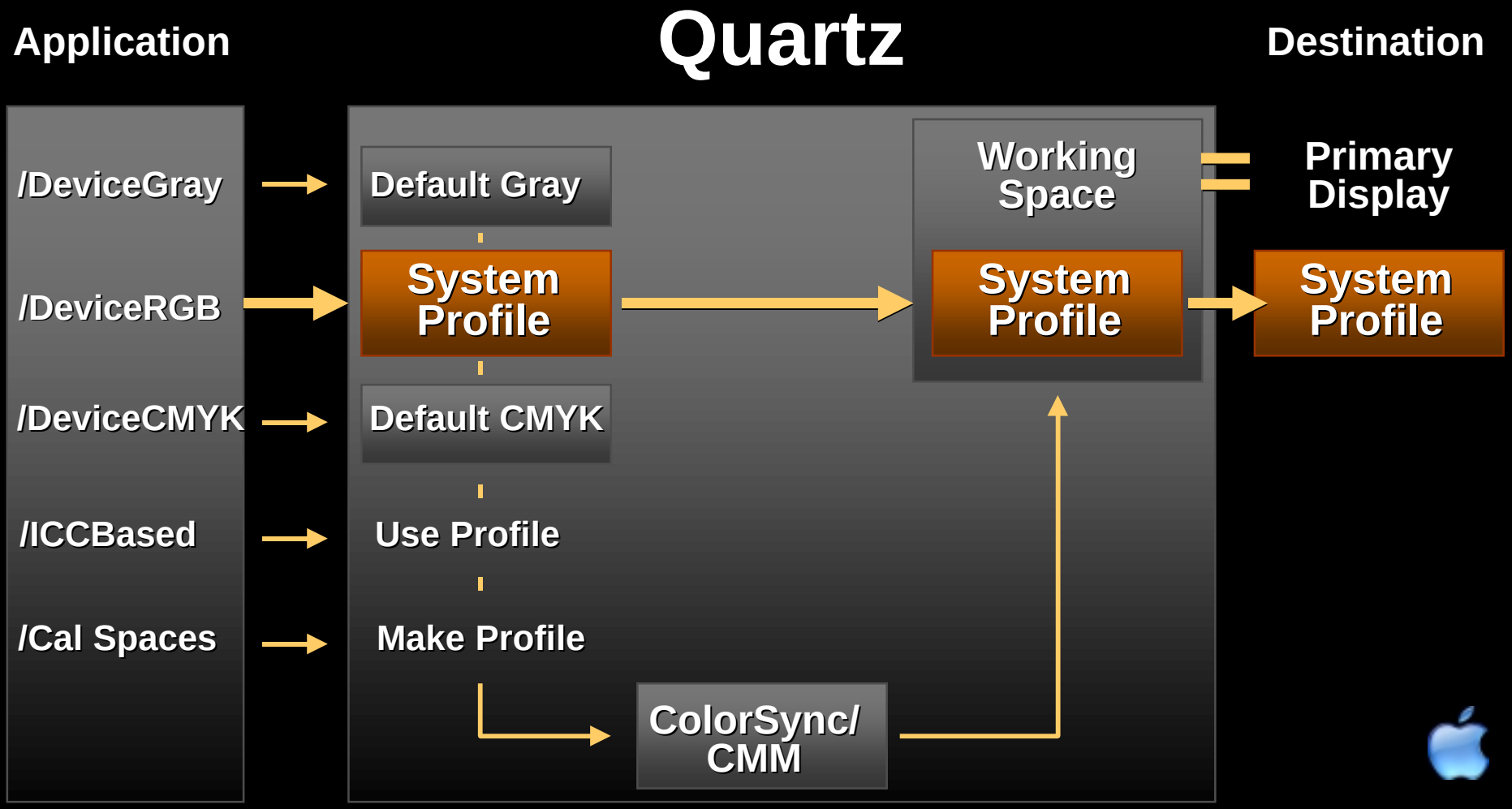
Quartz Default RGB = ICC RGB Profile

Quartz Default CMYK = CMYK to RGB → Quartz Default RGB

Color in Quartz Drawing Model

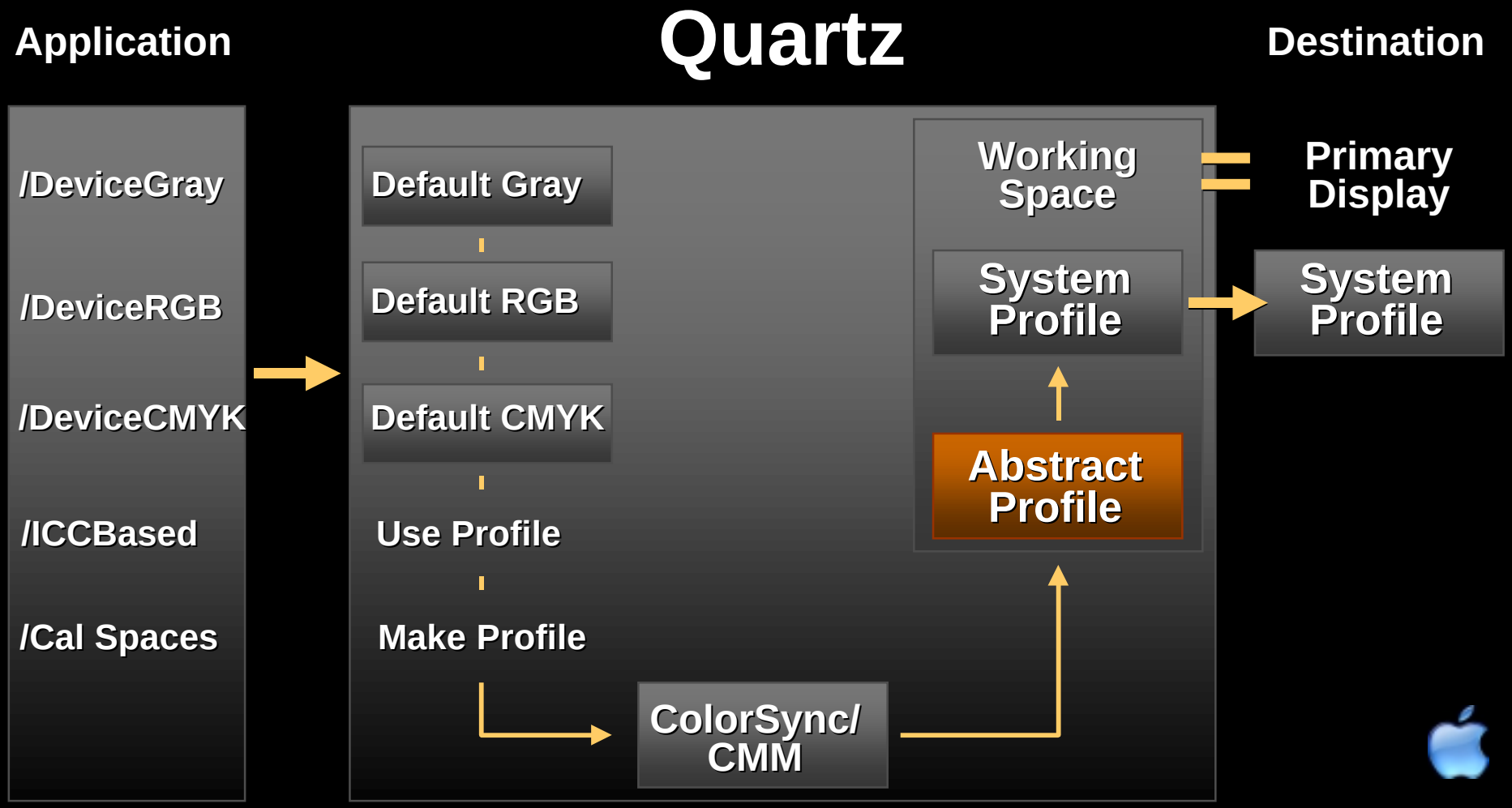


Color in Quartz Drawing Model



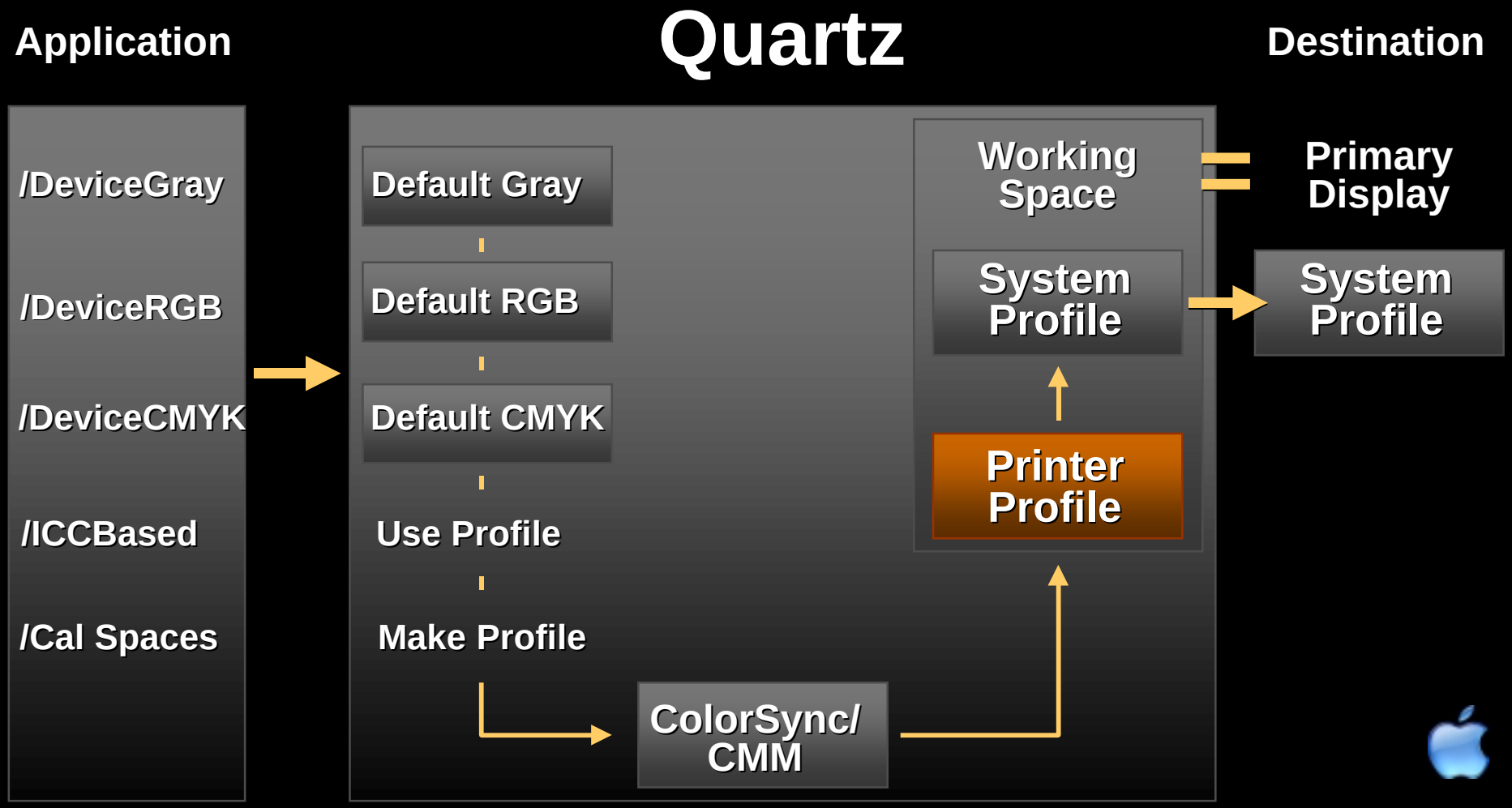
Color in Quartz Drawing Model

Applying Special Effects Abstract Profile



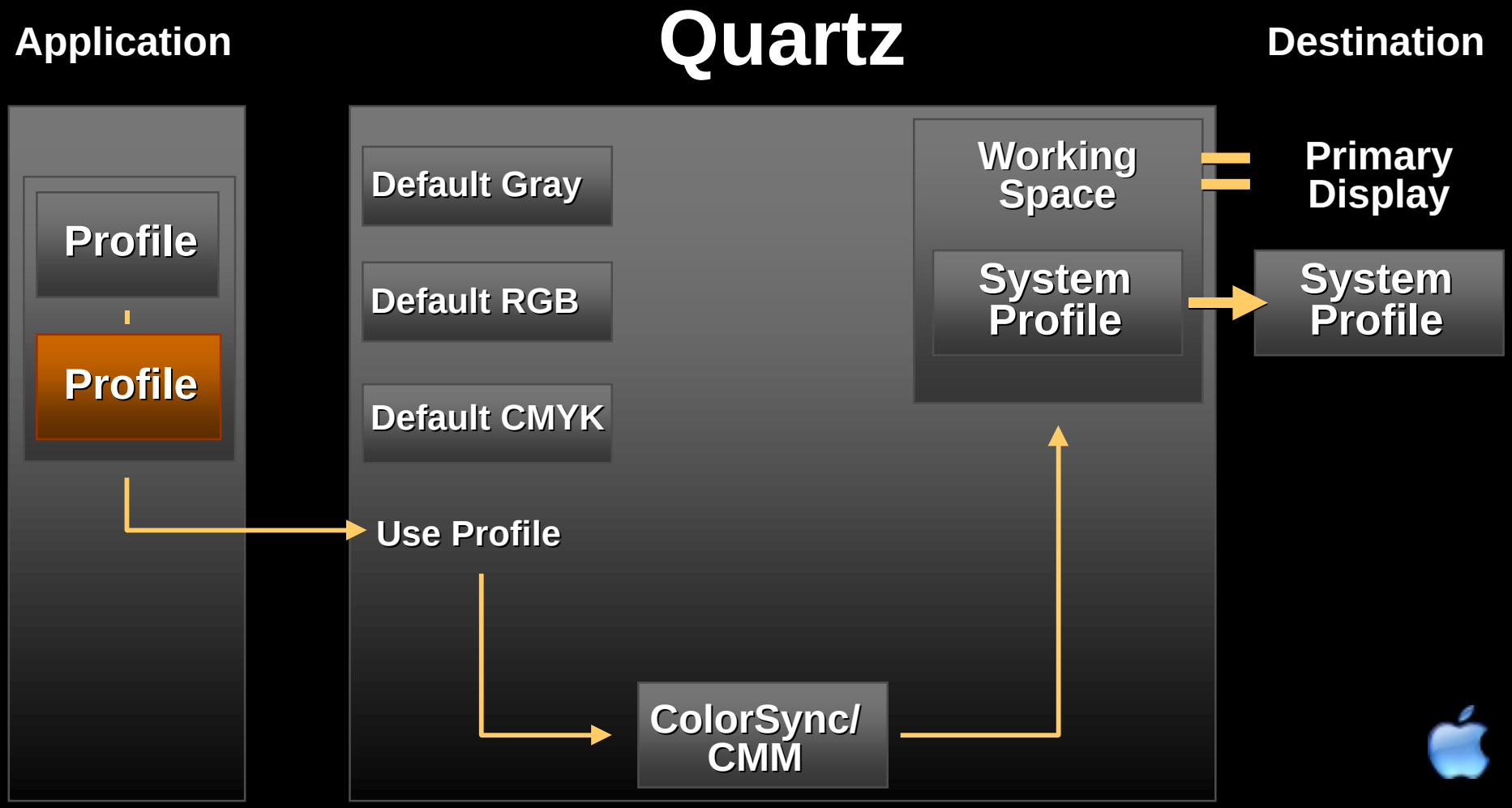
Color in Quartz Drawing Model

Softproofing Printout on Primary Display

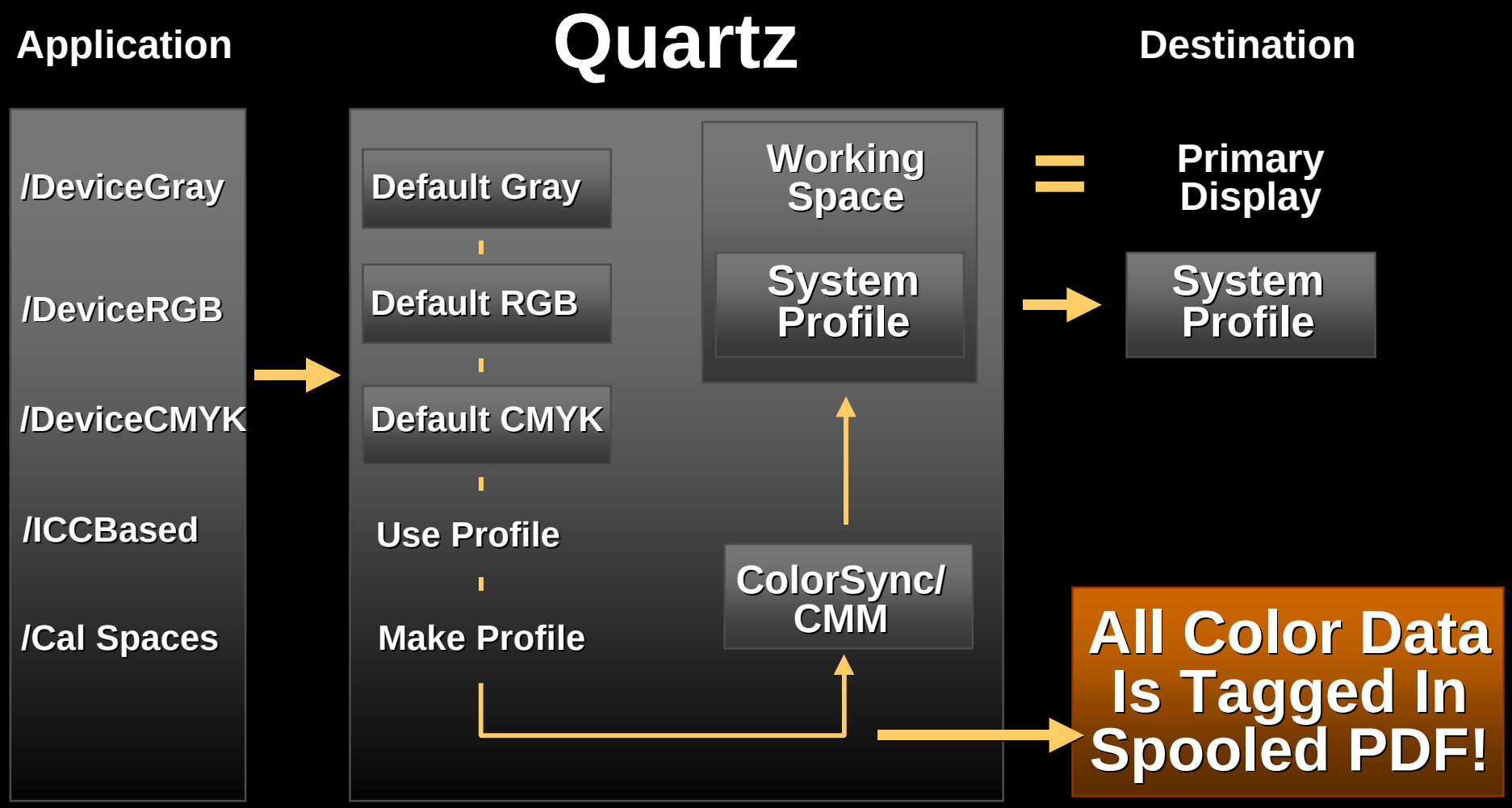


Color in Quartz Drawing Model

"Free transform" source space

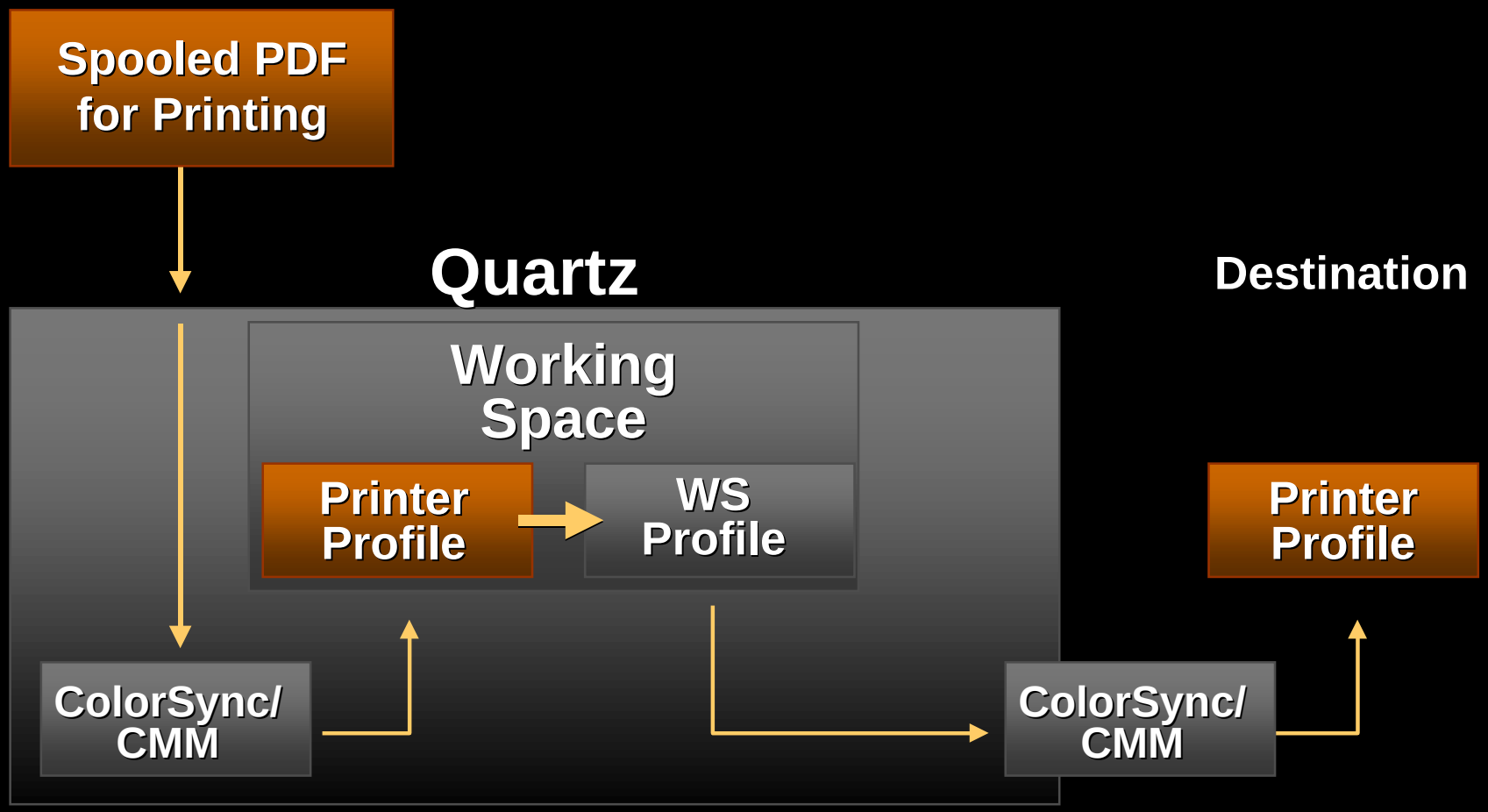


Quartz Printing Front End



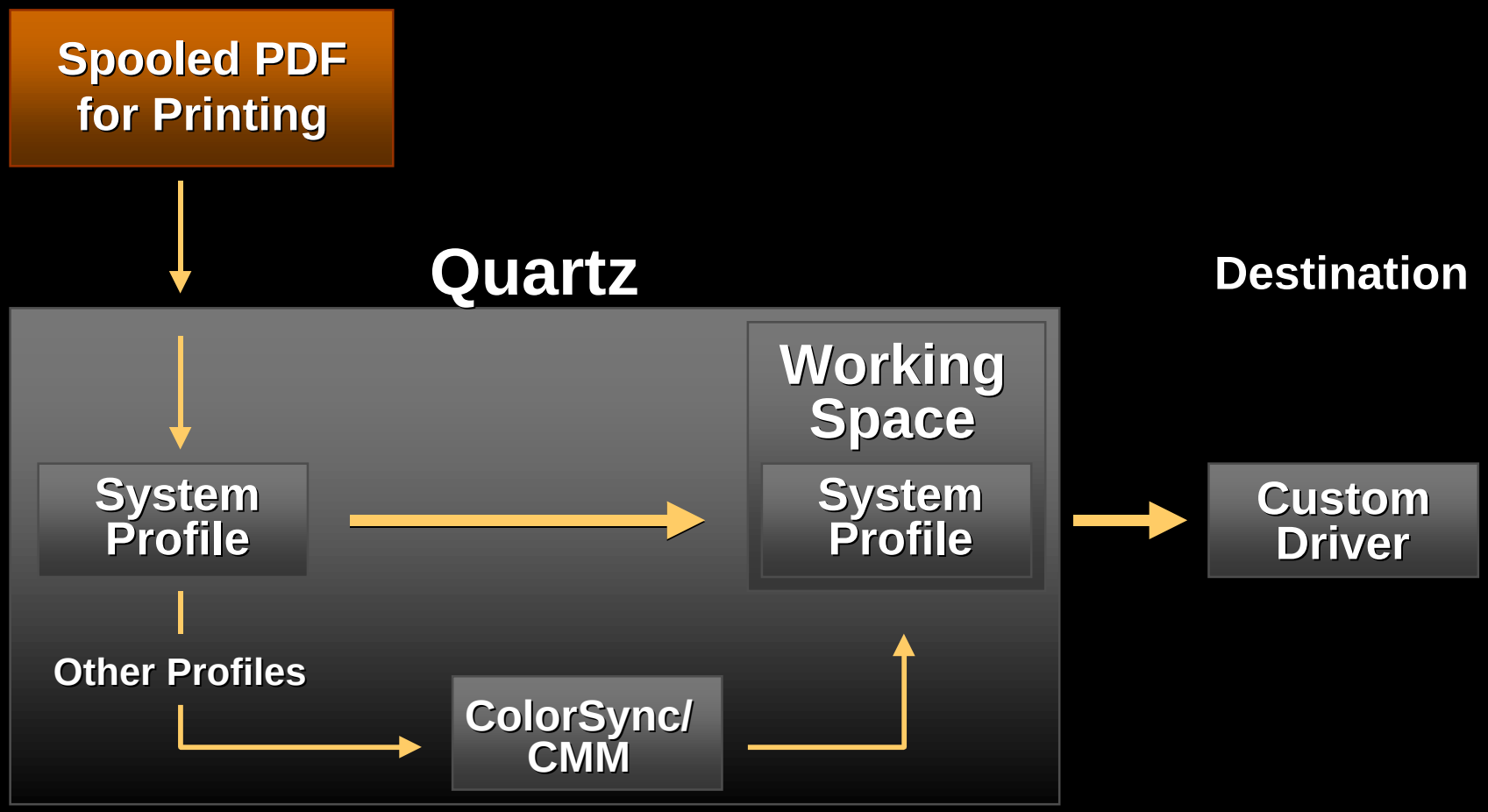
Quartz Raster Printing Back End

ColorSync Color Matching

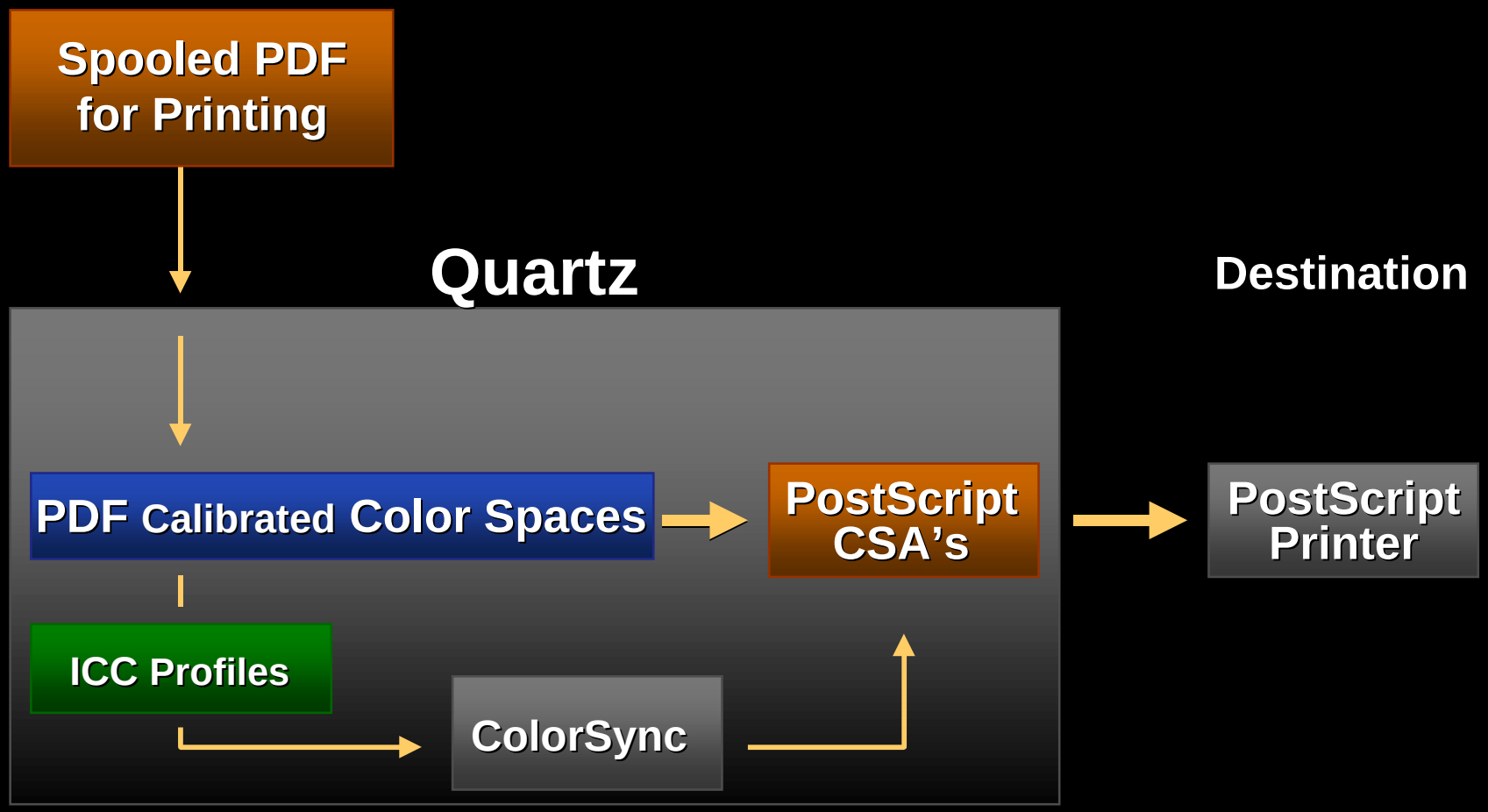


Quartz Raster Printing Back End

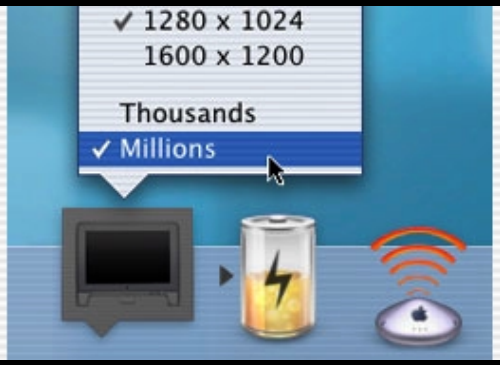
Custom Driver Color Matching



Quartz in PostScript Back End



Profiles on Mac OS X



Profiles on Mac OS X

- **CMGetColorSyncFolderSpec(vRefNum, …)**
- New Standard Locations
  - /System/Library/ColorSync/Profiles/ **kSystemDomain**
  - /Library/ColorSync/Profiles/ **kLocalDomain**
  - /Network/Library/ColorSync/Profiles/ **kNetworkDomain**
  - ~/Library/ColorSync/Profiles/ **kUserDomain**
  - Subfolders of all of the above (one level deep)
- Use **CMIterateColorSyncFolder** to find all profiles
- Classic profile folder is not searched. **kClassicDomain**

Profiles on Mac OS X

- Where do I install my profiles?
  - An application should install profiles in:
    /Library/ColorSync/Profiles/
    or in a subfolder of it
  - Users can install personal profiles in:
    ~/Library/ColorSync/Profiles/

Profiles on Mac OS X

- Profile prerequisites
  - Any type/creator (preferably prof/sync)
  - Any or no suffix (preferably .icc)
  - Value ‘acsp’ in header of file
  - A valid description tag
- Use Profile First Aid to make sure your profiles are free of common errors

Profiles on Mac OS X

- New optional profile tags
  - ‘mluc’ Multi-localized UniCode Tag
  - New API for easy modification of this tag:

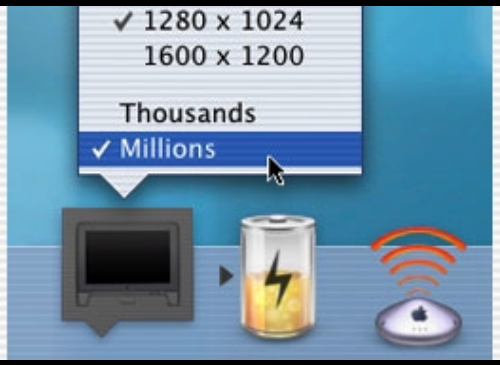
```
– CMError CMCopyProfileLocalizedStringDictionary (
        CMProfileRef prof,  OSType tag,  CFDictionaryRef* dict);
– CMError CMSetProfileLocalizedStringDictionary (
        CMProfileRef prof,  OSType tag,  CFDictionaryRef dict);
– CMError CMCopyProfileLocalizedString (
        CMProfileRef prof,  OSType tag, CFStringRef reqLocale,
        CFStringRef* locale,  CFStringRef* str);

– err = CMCopyProfileLocalizedString (
        prof, ‘dscm’, CFSTR(“enUS”), nil, &theStr);
```

ICC4 Profiles

- The new ICC 4.0 Profile Format Specification provides for more flexibility and power
  - New version and MD5 checksum in header
  - New tag data types (‘mluc’, ‘mBA ‘, ‘mAB ‘, ‘para’)
  - Some existing tags will contain new data formats
    - E.g. ‘A2B0’ may contain ‘mft1’ or ‘mAB ’ data.
- ColorSync and the Apple CMM will be ICC4 savvy
- If your app does not parse profiles, and simply uses profile references for color matching, everything will “just work”

CMMs on Mac OS X



CMMs on Mac OS X

- Standard locations:
  - /Library/ColorSync/CMMs
  - ~/Library/ColorSync/CMMs
- Use **CMMIterateCMMInfo** to find all CMMs
- CFBundle based
  - See CFBundle documentation for details
- Familiar entry points:
  - **CMMOpen, CMMConcat, CMMMatchBitmap, CMMClose**

CMMs on Mac OS X

- Apple CMM Features
  - Installed as default, tuned for integration with Quartz
  - Handles all profiles types and image spaces
- Developers of CMMs should thoroughly test their code in case the user sets it as the default

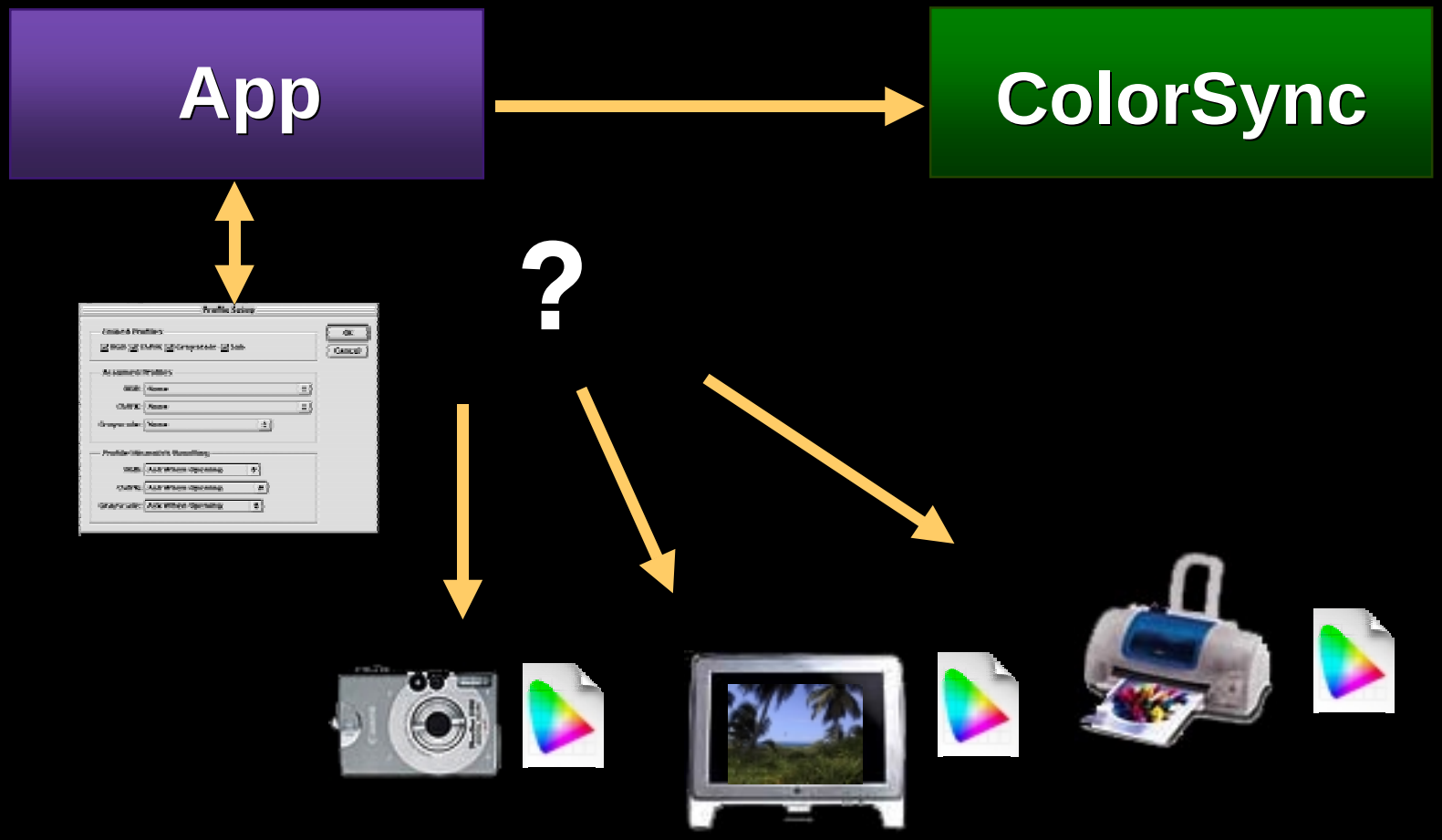
Demo

David Hayward
ColorSync Overview

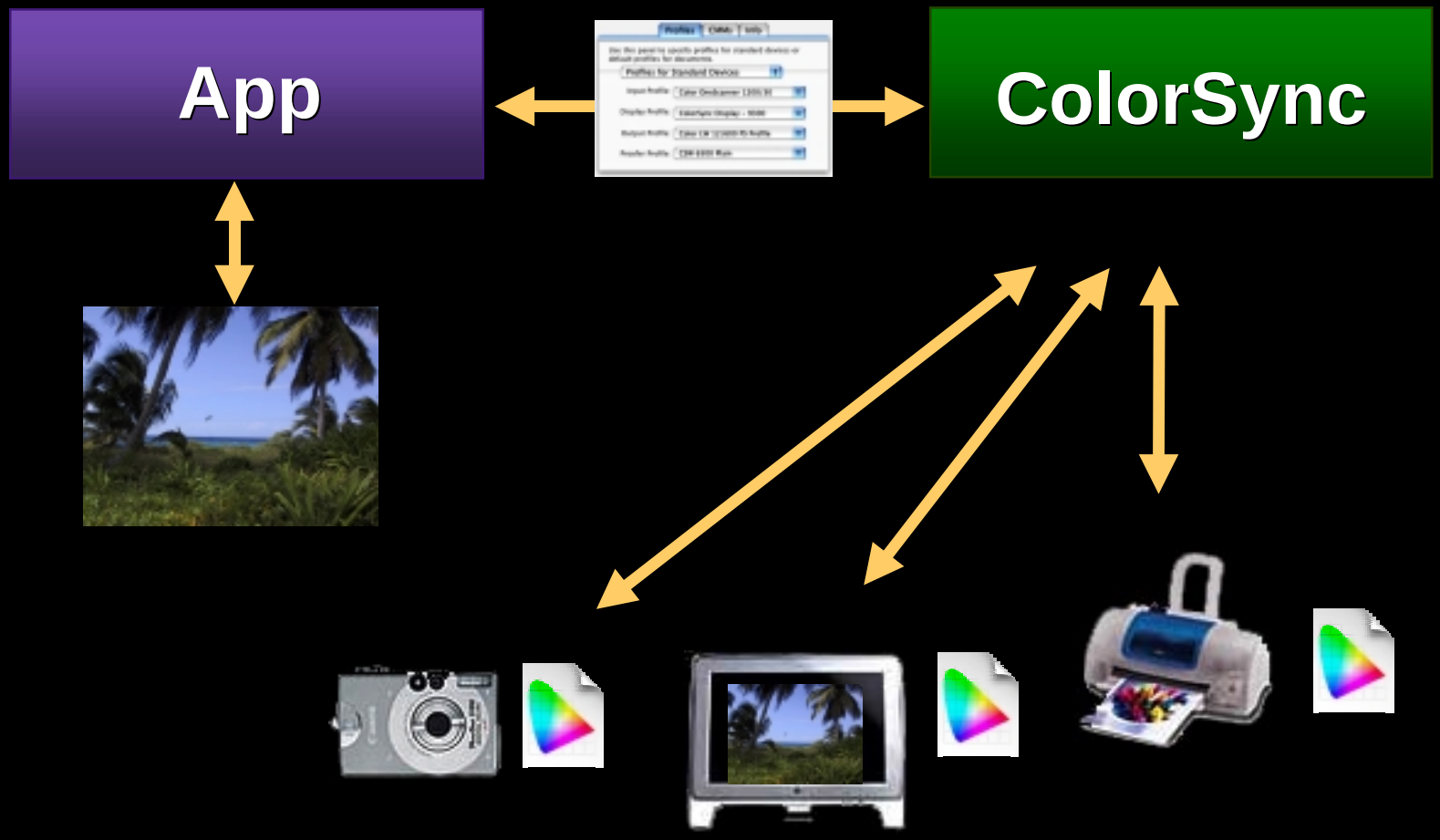
ColorSync Preferences and Supporting APIs

Eric Broadbent
ColorSync Engineer

ColorSync Preferences and Supporting APIs



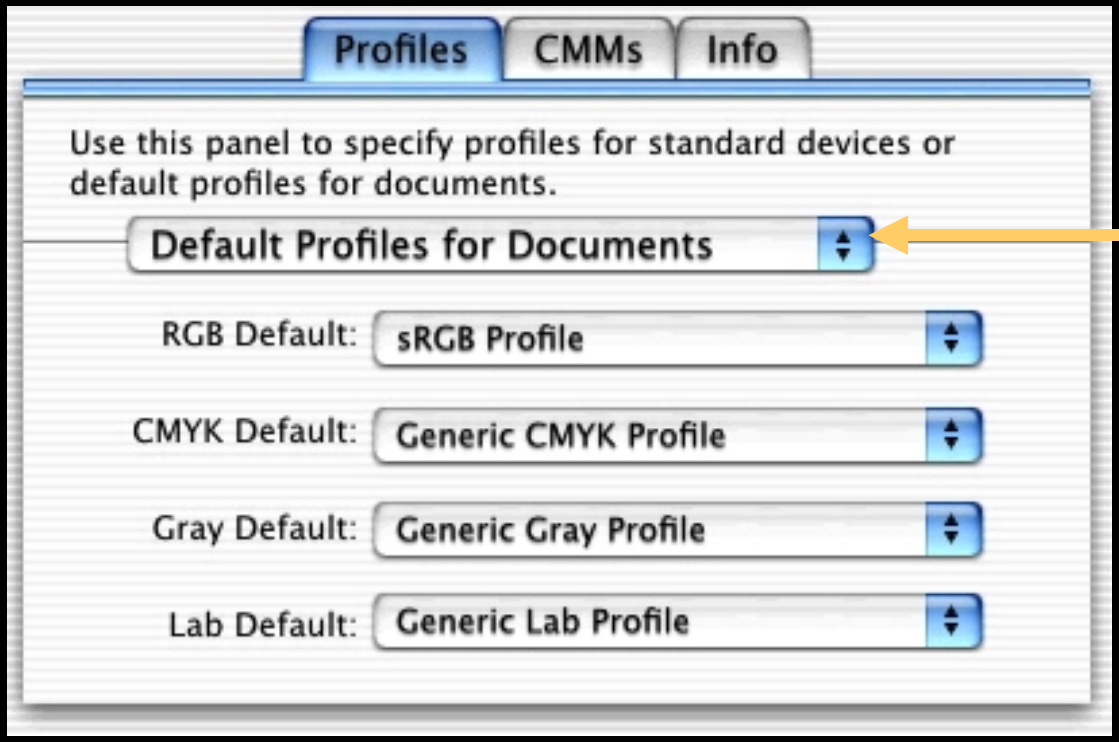
ColorSync Preferences and Supporting APIs



ColorSync Preferences and Supporting APIs

- ColorSync Preference Panel was re-designed in ColorSync 3.0 to:
  - Unify how and where users control color
  - Formalize common color management concepts (color devices, color spaces)
  - Provide an open and extensible architecture for future enhancements

Profiles for Standard Color Spaces



"Document" view

Supports document management:

docs may contain image data in different color spaces, with no source profiles

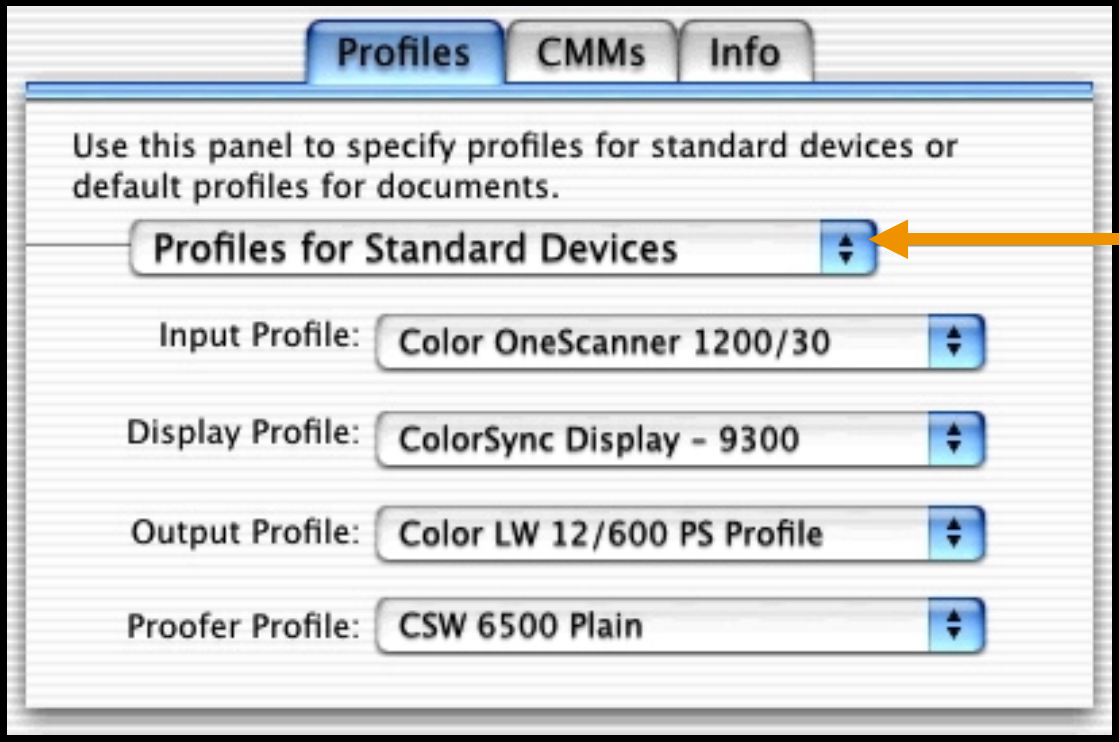
Profiles for Standard Color Spaces

- API For Access:

```
enum {
        cmXYZData       = 'XYZ ',
        cmLabData       = 'Lab ',
        cmRGBData       = 'RGB ',
        cmSRGBData      = 'sRGB',
        ... };

CMGetDefaultProfileBySpace (  OSType          space,
                              CMProfileRef*   prof );
```

Profiles for Standard Devices



"Device" view

Applications may want to apply some knowledge of what kind of device the color data is coming from or going to

Profiles for Standard Devices

- API For Access:

```
enum {
        cmInputUse      = 'inpt',
        cmOutputUse     = 'outp',
        cmDisplayUse    = 'dply',
        cmProofUse      = 'pruf'  };

CMGetDefaultProfileByUse (  OSType          use,
                            CMProfileRef*   prof );
```

ColorSync Prefs Code Sample

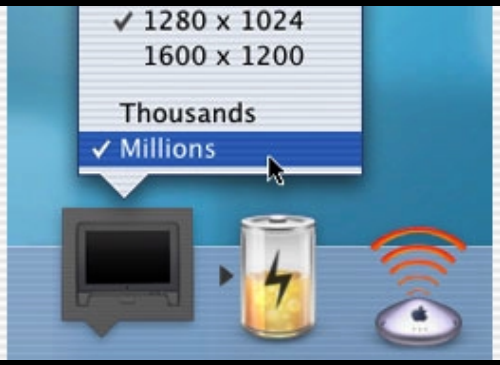
```
CMError  myMatchImage  (myImageRef    image)
{
        CMProfileRef        source, dest;
        CMWorldRef          cw;
        CMBitMap            bm;

        source = myGetEmbeddedProfile (image);
        if (source == nil)
        {
                space = myGetColorImageSpace (image);
                err = CMGetDefaultProfileBySpace (space, &source);
        }
        err = CMGetDefaultProfileByUse (cmDisplayUse, &dest);
        err = NCWNewColorWorld (&cw, source, dest);
        bm = myGetImageBitMap (image);
        err = CWMatchBitMap (cw, &bm, ..., ..., ...);
        ...
```

ColorSync Preferences Summary

- Since all data is color managed, your App can participate simply by becoming more aware of “from” and “to”
- Most applications can simply use these APIs for color management, to get user’s preferences
- Profile Accessor APIs are extensible, a bridge to future device and color-space support
- Applications can manage color from both document-centric and device-centric perspectives

ColorSync Device Support on Mac OS X



ColorSync Device Support

- On Mac OS X , we have new device managers for input, display, and printing
- Opportunity to integrate with ColorSync and provide both awareness of devices and access to their profiles
- Implemented with the following services:
  - Device registration
  - Profile registration
  - Default Device & Default Profile accessors
  - Notification

ColorSync Device Support

- Device & Profile Registration:
  - Device Mgrs detect presence of devices
  - Can also discover or build device profiles
- ColorSync APIs to capture device & profile info:

CMRegisterColorDevice
CMUnregisterColorDevice
CMSetDeviceFactoryProfiles

ColorSync Device Support

- Default Devices and Default Profiles:
  - APIs provide access to defaults, not just any device profile
  - When profiles are registered, one is identified as the default
  - When a user changes settings, the default device or default profile may change
  - Device Managers keep track of which device is default

- APIs to manage default devices and profiles:

  CMSetDefaultDevice

  CMSetDeviceDefaultProfileID

ColorSync Device Support

- Calibration Support:
  - Profiles are registered with ID, “mode” name (e.g. “plain paper”)
  - Profiles are referenced during processing by ID
  - Factory profiles can be “overlaid” with custom profiles
  - A calibration application can get the factory profile for a given mode, and then set a new profile (by ID) to be used for that mode

- APIs supporting profile customization:

  CMGetDeviceFactoryProfiles

  CMSetDeviceProfiles

  CMSetDeviceProfile

ColorSync Device Support

- Notifications:
  - ColorSync posts notifications to a “distributed” center, so any process on the system can register to get them
  - Notifications for:
    - Changes to the default device for a device class
    - Changes to a devices factory or custom profiles
    - Changes to a device’s default profile
    - Device registration/unregistration
- APIs (Non-ColorSync) supporting notification:

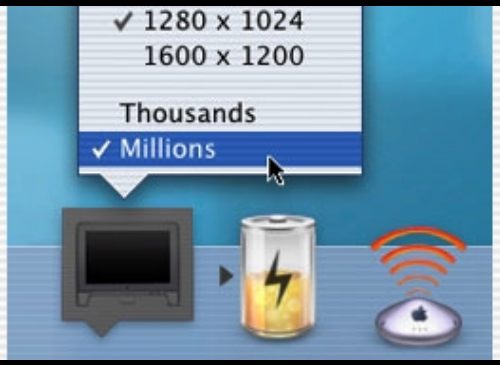
  CFNotificationCenterAddObserver

  NSDistributedNotificationCenter

ColorSync Device Support Summary

- ColorSync is now integrated with Device Mgrs
- Applications have access to default profile for standard devices (input, display, output)
- Applications can get more specific profile information for registered devices
- Calibration supported via profile access, abstraction, customization
- Notification Services

Demo



David Hayward
ColorSync Device Integration
On Mac OS X

For More Information

- www.apple.com/colorsync
- www.apple.com/developer/sdk/
- colorsync-dev and colorsync-users lists
- colorsync@apple.com

Roadmap

108 PDF, Quartz, and Mac OS X: Learn about Mac OS X's support for PDF	Room J2 **Tues., 9:00am**
107 2D Graphics Using Quartz: An Introduction to Quartz 2D	Room A1 **Tues., 10:30am**
110 Drawing Unicode Text with ATSUI: Learn how to draw Unicode text on Mac OS X	Room J2 **Tues., 3:30pm**
117 Image Capture: System-level support for Digital Cameras	Room C **Wed., 10:30am**

http://developer.apple.com/events/urls.html

Roadmap

118 ColorSync: The color engine of Mac OS X	Room C **Wed., 2:00pm**
113 Text on Mac OS X: All about DrawThemeText, Unicode Edit Text Control, and MLTE	Hall 2 **Wed., 3:30pm**
404 OpenGL: High Performance 2D Blazingly fast 2D graphics using OpenGL	Room J2 **Wed., 3:30 pm**
124 Printing: Using Mac OS X's print architecture	Room A2 **Thurs., 3:30pm**

http://developer.apple.com/events/urls.html

Roadmap

132 G&I Tips and Tricks: Optimize your graphics and printing	Room A2 **Fri., 9:00am**
FF118 Feedback Forum: Graphics Give us your feedback on Mac OS X graphics	Room J1 **Fri., 2:00pm**

http://developer.apple.com/events/urls.html

Who to Contact

Travis W. Brown
Mac OS X Imaging Technology Manager
Apple Worldwide Developer Relations
travisb@apple.com

John Zimmerer
ColorSync Product Manager
johnz@apple.com

http://developer.apple.com/events/urls.html

Q&A

Travis W. Brown
Mac OS X Imaging Technology Manager
Apple Worldwide Developer Relations
travisb@apple.com

WWDC

Worldwide Developers Conference 2001